

REAL PARTY IN INTEREST

The real party in interest in the present Appeal is International Business Machines Corporation, the Assignee of the present application as evidenced by the Assignment recorded at reel 010282 and frame 0412.

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellant, the Appellant's legal representative, or assignee, which directly affect or would be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-33 stand finally rejected by the Examiner as noted in the Final Office Action dated October 4, 2003 and the Advisory Action dated December 23, 2003.

STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the final rejection that leads to this appeal.

SUMMARY OF THE INVENTION

As described within the Specification on pages 4 and 5, Appellant's invention provides a method for automatically updating a file downloaded to a client system from a network server (e.g., an Internet server) with a new version of the file currently provided on the network server. A source descriptor is attached to a locally (client) saved copy of the file and provides routing address of the server from which the file was downloaded. When the file is opened on the local/client system, the source descriptor is utilized to access the server, and the server is checked for current updates or a newer version of the downloaded file. The client-stored file is updated/replaced whenever there is a newer version of the file on the network server. The source descriptor is also attached to the newly downloaded version of the file for future update operations.

In one embodiment, the user is alerted to the presence of a newer version of the file and prompted whether to replace the local file with the newer server version. If the user elects to download the newer version, the local file is replaced with the newer, sever version at the

local/client system. In one embodiment, a time interval is established for determining when to check the server for updates.

Appellant's exemplary claim recites: "evaluating **at said client** ... to determine if a **source identifier** is present in said downloaded file; checking said source periodically utilizing said source identifier to **determine if a newer version** of said downloaded file **exists**; and **replacing** at said client, in response to the presence of said newer version of said downloaded file, said **downloaded file with said newer version**" (emphases added).

ISSUES

The primary issue for appeal is whether Examiner's rejections of Appellant's claims under 35 U.S.C. § 103(a), as being unpatentable over Smith, et al. (U.S. Patent No. 6,006,206) and over Smith in view of Kullick, et al. (U.S. Patent No. 5,764,992) are well founded. Tantamount to a resolution of that issue is a determination whether Smith or the combination of Smith and Kullick suggests to one skilled in the art a client-side "file-updating" operation in which a source of a downloaded file is periodically checked for a newer version of the downloaded file and then the downloaded file is replaced with the newer version at the client.

GROUPING OF THE CLAIMS

For purposes of this Appeal, all claims stand or fall together as a single group.

ARGUMENT

Examiner's rejection of Appellant's exemplary Claim 1 as being unpatentable over Smith is not well founded and should be reversed.

Appellant hereby incorporates by reference the arguments proffered in Amendment A filed on July 21, 2003 and Response B filed on December 4, 2003 with respect to both 103 rejections. As stated within these incorporated arguments, one skilled in the art would not find Appellant's claimed invention obvious over Smith because Smith fails to suggest the subject matter recited in Appellant's exemplary claim. Specifically, Smith fails to suggest several functional features of Appellant's claims. Key among these features is the replacement at a client of a previously downloaded file with a newer version of the file when the client determines

that the newer version is currently available at the server that sourced the downloaded file (paraphrased).

Notably, Smith also fails to suggest other features of Appellant's claimed invention, including: (1) evaluating at a client if a source identifier is present in the downloaded file and attaching (at the client) a source identifier if no source identifier is present in the downloaded file; (2) actually "prompting" a user to replace the downloaded file with a newer version when a newer version exists; (3) defining a default, automatic time interval at which a check is made from the client for a newer version of the downloaded file and enabling a user to adjust the time interval; and (4) checking the source whenever the downloaded file is opened, and, when there is a defined periodic time interval at which an automatic checking occurs, overriding the time interval by initiating the check at the time the downloaded file is opened.

Smith provides a status code generator that checks the health of a data source and then "automatically updates the status code when the operating status of a corresponding data source changes" (emphasis added) (Abstract). Smith's use of the terms "status code" and "operating status" relate solely to the "health" (i.e., ability to provide and transmit data, etc.) of the source (or server) providing data. Smith's focus is primarily on the source's overall ability to transmit data from the source. Smith is not focused on and does not even contemplate downloading to the client particular files that may occasionally/periodically be updated at the server. *See, for example*, Figure 5, which clearly illustrates that the source status signal is related to the operation of the source (e.g., the server providing data) and is at the network/server function (steps 501–504) rather than a client function.

According to Smith, the client system merely receives "the status codes, which it processes to determine whether there is a problem in the receipt of the financial data, which prevents the terminal's receipt of the data in real time" (emphasis added) (Abstract; col. 3, lines 30-37, 40-42, etc.). Smith displays a stale copy of the data when the data cannot be received in real time because of the problem identified by the status code.

One skilled in the art recognizes that the status code indicates hardware and/or communication/transmission problems at the source. Thus, Smith, in effect, determines when a technical problem (e.g., faulty data transmission) exists at the data source and responds by sending the status code to allow the display of stale data currently on the client system whenever such a problem is identified by the status code. Displaying old data because a status code is received indicating there is a problem with a source of the data that prevents new data from being sent from the source to the client is NOT synonymous with nor suggestive of determining when there is a newer version of a file on a network server and downloading the newer version to the client to replace the old version stored on the client. Smith clearly does not suggest automatically responding at the client to updates to the actual file on the server based on a client-issued request for a download of the updated version, if such an updated version is available.

Examiner's analysis of what features of Smith are suggestive of Appellant's claims includes several inconsistencies and fails to account for several key inventive features that distinguishes Appellant's claims from Smith. For example, as described by Examiner and recited by Smith, col. 3, lines 20-34 provides a description of a status code generator (at the source) to update the status code covering the "operating status" of the data source. There is absolutely no suggestion within that description of updating a previously downloaded file. Further, Smith describes transmitting an "updated status code" only. The operational/functional features of a status code generator (for hardware and/or transmission status checking and signaling via a status code) are not suggestive of checking a server/source to determine when/if a newer version of a downloaded file exists on the server and actually downloading the newer version to a client system to replace the previously downloaded version.

As another example, Examiner references col. 12, lines 33-67 and Figures 5 and 6 (and descriptions thereof), which describes a data health monitor for detecting and identifying non-current financial data. This section clearly describes detecting outdated data versus Appellant's claims which are directed to identifying new/updated files. Further, with Smith, the client side terminal may select stale or real-time display mode "based upon ... status code and updated status code." As previously mentioned, displaying stale data based on the value of a status code and updated status code is a completely different application and addresses a very different

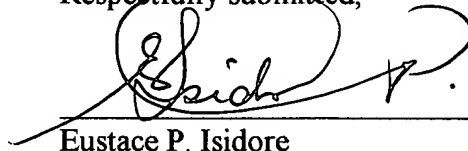
problem from what Appellant's claims are designed to address. Appellant's claims are clearly directed to providing an updated copy of a file to a client whenever a new/updated file is provided/available at the server.

With the above arguments, Appellant has clearly shown that Smith does not contemplate or suggest the various features that are recited by Appellant's exemplary claim. Further, one skilled in the art would not find Appellant's invention obvious in light of Smith. Appellant's claims are therefore not unpatentable over Smith and should be allowed. For those reasons, Examiner's rejection of Appellant's claims is not well founded and should be reversed.

CONCLUSION

Appellant has pointed out with specificity the manifest error in the Examiner's rejections and the claim language which renders the invention patentable over the cited reference(s). Appellant, therefore, respectfully requests that this case be remanded to the Examiner with instructions to issue a Notice of Allowance with respect to all pending claims.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "E. Isidore", is written over a horizontal line.

Eustace P. Isidore

Registered with Limited Recognition (see attached)

DILLON & YUDELL LLP

P.O. Box 201720

Austin, Texas 78720-1720

512.343.6116

ATTORNEY FOR APPELLANT(S)

APPENDIX

1. A method for keeping files current for use in a client computer system coupled to a network, the method comprising the steps of:

evaluating at said client a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file;

checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and

replacing at said client, in response to the presence of said newer version of said downloaded file, said downloaded file with said newer version.

2. The method as recited in claim 1 wherein said step of evaluating further includes the step of attaching, in response to no source identifier being present, a source identifier to said downloaded file.

3. The method as recited in Claim 1 wherein said step of replacing said downloaded file includes the steps of:

providing an indication to a user that said newer version of said file exists;

prompting said user to replace said downloaded file with said newer version; and

replacing, in response to said user requesting said newer version, said downloaded file with said newer version.

4. The method as recited in Claim 1 wherein said source identifier is located in the extended attribute of said downloaded file.

5. The method as recited in Claim 1 wherein said downloaded file comprises one or more stored parameters from among: (1) a signature string utilized to find said source identifier within said file; (2) a locator string identifying a location from which the file is sourced; (3) a date/time and version number of said file; and (4) a checksum string covering prior entries of said file.

6. The method as recited in Claim 1 wherein said source identifier is an uniform resource locator (URL).
7. The method as recited in Claim 1 wherein said step of checking said source periodically includes:
 - defining a default, automatic time interval at which said checking step is initiated; and
 - enabling a user to adjust said time interval, if desired.
8. The method as recited in Claim 1, wherein said replacing step further comprises:
 - renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and
 - storing said newer version of said downloaded file with the current working name of the downloaded file.
9. The method as recited in Claim 1 wherein said step of checking said source comprises checking said source whenever said downloaded file is opened, wherein, when said checking step includes a defined periodic time interval at which said checking is automatically initiated, said method further comprises overriding said time interval by initiating said checking step whenever said downloaded file is opened.
10. The method as recited in Claim 1, further comprising storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.
11. The method as recited in Claim 1 wherein said network is a packet network.
12. A computer system for use in a network environment, comprising:
 - a processor;
 - an update manager coupled to said processor, including:

means for evaluating a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file;

means for checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and

means for replacing, in response to the presence of a newer version of said downloaded file, said downloaded file with said newer version.

13. The computer system as recited in Claim 12 wherein said means for evaluating further includes means for attaching, in response to no source identifier being present, a source identifier to said downloaded file.

14. The computer system as recited in Claim 12 wherein said means for replacing said downloaded file includes:

means for providing an indication to a user that said newer version of said file exists;

means for prompting said user to replace said downloaded file with said newer version;
and

means for replacing, in response to said user requesting said newer version, said downloaded file with said newer version.

15. The computer system as recited in Claim 12 wherein said source identifier is located in the extended attribute of said downloaded file.

16. The computer system as recited in Claim 12 wherein said downloaded file comprises one or more stored parameters from among: (1) a signature string utilized to find said source identifier within said file; (2) a locator string identifying a location from which the file is sourced; (3) a date/time and version number of said file; and (4) a checksum string covering prior entries of said file.

17. The computer system as recited in Claim 12 wherein said source identifier is an uniform resource locator (URL).

18. The computer system as recited in Claim 12 wherein said means for checking said source periodically includes:

defining a default, automatic time interval at which said checking step is initiated; and
enabling a user to adjust said time interval, if desired.

19. The computer system as recited in Claim 18 wherein said replacing step further comprises:

renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and

storing said newer version of said downloaded file with the current working name of the downloaded file.

20. The computer system as recited in Claim 12 wherein said means for checking said source comprises checking said source whenever said downloaded file is opened, wherein, when said checking includes a defined periodic time interval at which said checking is automatically initiated, said system further comprises means for overriding said time interval by initiating said checking whenever said downloaded file is opened.

21. The computer system as recited in Claim 12, further comprising means for storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.

22. The computer system as recited in Claim 12 wherein said network is a packet network and said computer system is a client system coupled to said network.

23. A computer program product comprising:

a computer-readable medium having stored thereon computer executable instructions for implementing a method for keeping files current for use in a client computer system coupled to a network, said computer executable instructions when executed, perform the steps of:

evaluating at said client a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file;

checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and

replacing at said client, in response to the presence of said newer version of said downloaded file, said downloaded file with said newer version.

24. The computer program product as recited in Claim 23 wherein said step of evaluating further includes the step of attaching, in response to no source identifier being present, a source identifier to said downloaded file.

25. The computer program product as recited in Claim 23 wherein said step of replacing said downloaded file includes the steps of:

providing an indication to a user that said newer version of said file exists;

prompting said user to replace said downloaded file with said newer version; and

replacing, in response to said user requesting said newer version, said downloaded file with said newer version.

26. The computer program product as recited in Claim 23 wherein said source identifier is located in the extended attribute of said downloaded file.

27. The computer program product as recited in Claim 23 wherein said downloaded file comprises one or more stored parameters from among: (1) a signature string utilized to find said source identifier within said file; (2) a locator string identifying a location from which the file is sourced; (3) a date/time and version number of said file; and (4) a checksum string covering prior entries of said file.

28. The computer program product as recited in Claim 23 wherein said source identifier is an uniform resource locator (URL).

29. The computer program product as recited in Claim 23 wherein said step of checking said source periodically includes:

defining a default, automatic time interval at which said checking step is initiated; and
enabling a user to adjust said time interval, if desired.

30. The computer program product as recited in Claim 29 wherein said replacing step further comprises:

renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and

storing said newer version of said downloaded file with the current working name of the downloaded file.

31. The computer program product as recited in claim 23 wherein said step of checking said URL comprises checking said source whenever said downloaded file is opened, wherein, when said checking step includes a defined periodic time interval at which said checking is automatically initiated, said method further comprises overriding said time interval by initiating said checking step whenever said downloaded file is opened:

32. The computer program product as recited in Claim 23, further comprising storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.

33. The computer program product as recited in Claim 23 wherein said network is a packet network.




**BEFORE THE OFFICE OF ENROLLMENT AND DISCIPLINE
UNITED STATES PATENT AND TRADEMARK OFFICE**

LIMITED RECOGNITION UNDER 37 CFR § 10.9(b)

Mr. Eustace P. Isadore is hereby given limited recognition under 37 CFR §10.9(b) as an employee of Dillon & Yudell L.L.P. to prepare and prosecute patent applications wherein the patent applicant is the client of Dillon & Yudell L.L.P., and the attorney or agent of record in the applications is a registered practitioner who is a member of Dillon & Yudell L.L.P. This limited recognition shall expire on the date appearing below, or when whichever of the following events first occurs prior to the date appearing below: (i) Mr. Eustace P. Isadore ceases to lawfully reside in the United States, (ii) Mr. Eustace P. Isadore's employment with Dillon & Yudell L.L.P. ceases or is terminated, or (iii) Mr. Eustace P. Isadore's current Employment Authorization card expires.

This document constitutes proof of such recognition. The original of this document is on file in the Office of Enrollment and Discipline of the U.S. Patent and Trademark Office.

Expires: June 26, 2004


Harry I. Moatz
Director of Enrollment and Discipline